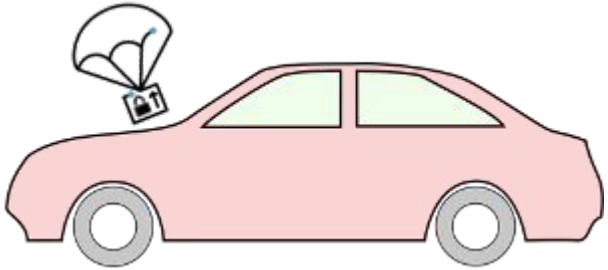
Uptane

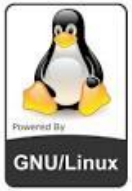
Securing Over-the-Air Updates
Against Nation State Actors



Justin Cappos
New York University
uptane.github.io

NYU TANDON SCHOOL OF ENGINEERING

What do these companies have in common?



What do these companies have in common?





Users attacked via software updater!

Windows

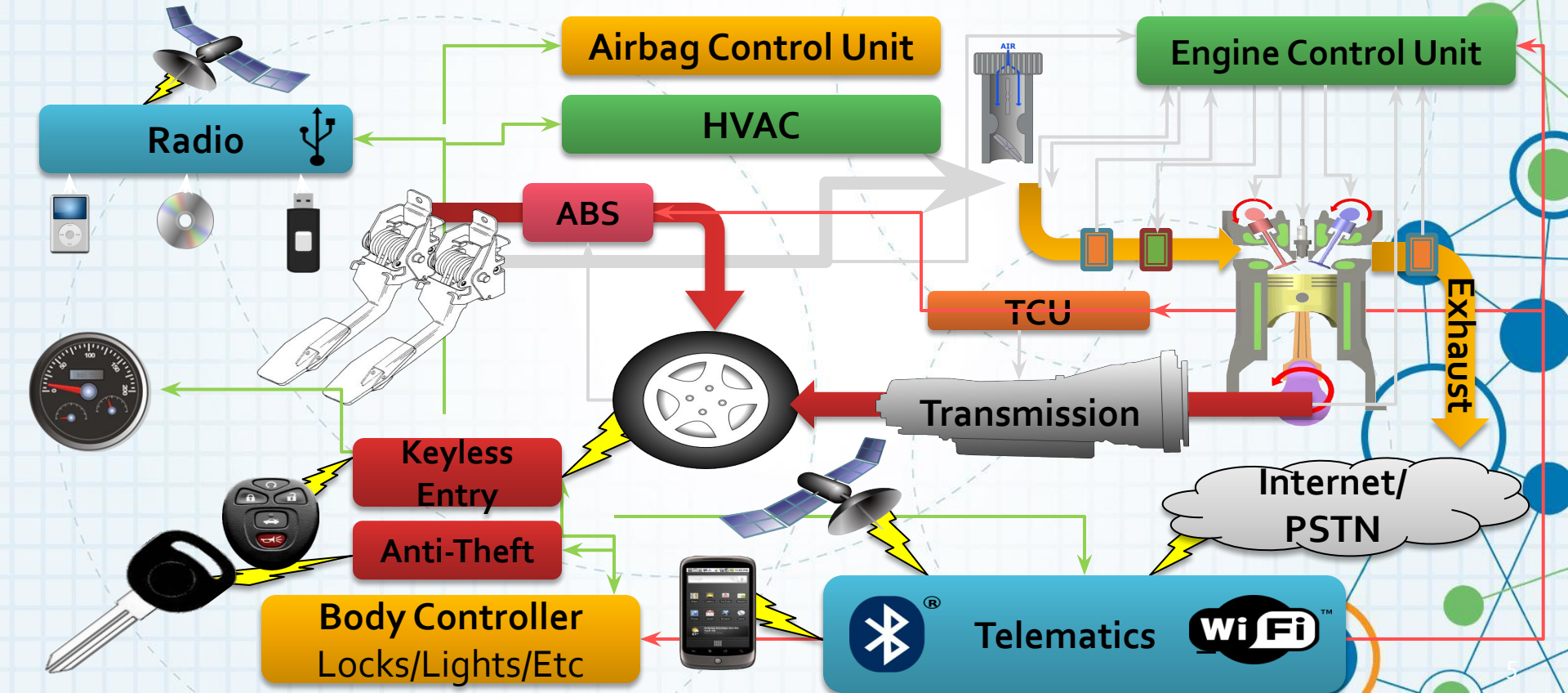
sourceforge

GitHub

Software repository compromise impact

- SourceForge mirror distributed **malware**.
- Attackers impersonate Microsoft Windows Update to spread Flame **malware**.
- Attacks on software updaters have massive impact
  - E.g. South Korea faced 765 million dollars in damages.
- NotPetya spread via software updates!

The modern automobile



Cars Are Dangerous

- Researchers have made some scary attacks against vehicles
  - remotely controlling a car's brakes and steering while it's driving
  - spontaneously applying the parking brake at speed
  - turning off the transmission
  - locking driver in the car

Cars are multi-ton, fast-moving weapons

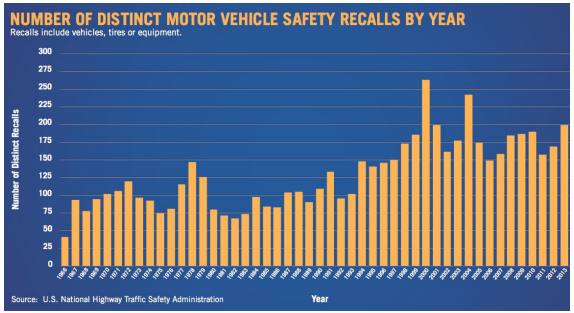
People will die

Updates Are Inevitable

- Millions of lines of code means bugs
- Regulations change -> firmware must change
- Maps change
- Add new features
- Close security holes
- Cars move across borders…

Updates Must Be Practical

- Updating software/firmware has often meant recalls.
- Recalls are extremely expensive
  - GM spent $4.1 billion on recalls in 2014
  - GM's net income for 2014 was < $4 billion
  - People do not like recalls.
- Updates must be over the air.



Updates Are Dangerous

- Update -> Control

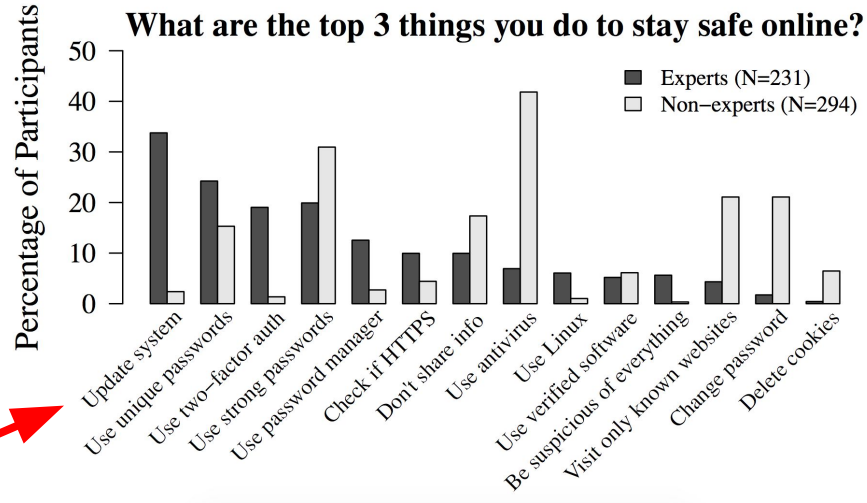
Secure Updates

- Nation-state actors pull off complex attacks
  - Must not have a single point of failure

What to do?

Must update to fix security issues

Insecure update mechanism is a new security problem

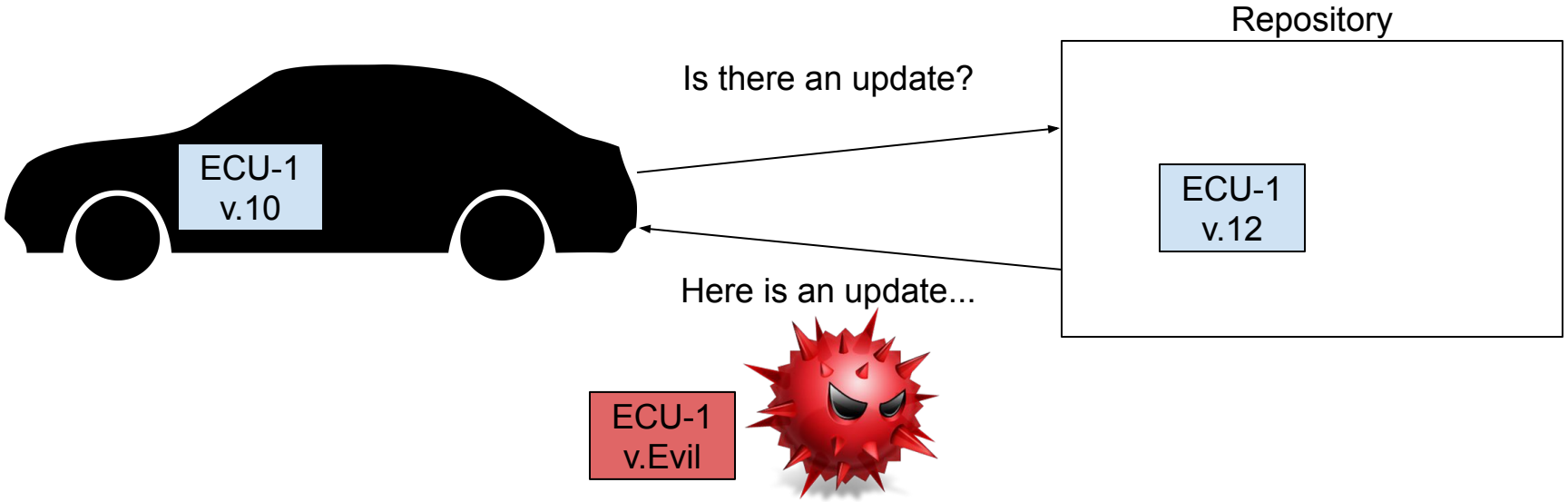


"...No one Can Hack My Mind": Comparing Expert and Non-Expert Security Practices
Ion, et al. SOUPS 2015

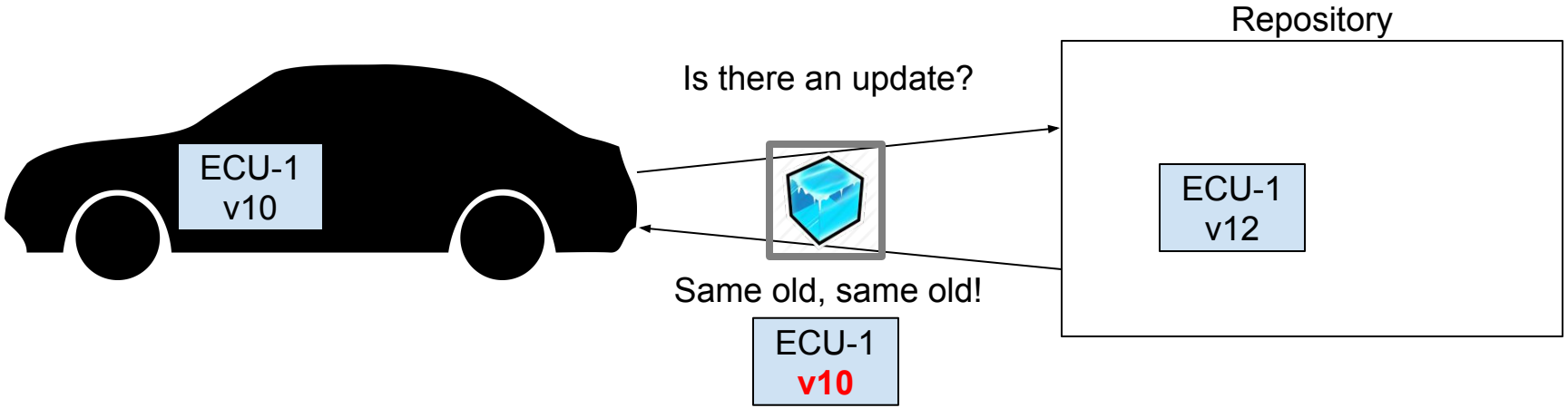
Attacks

What are some of the attacks?

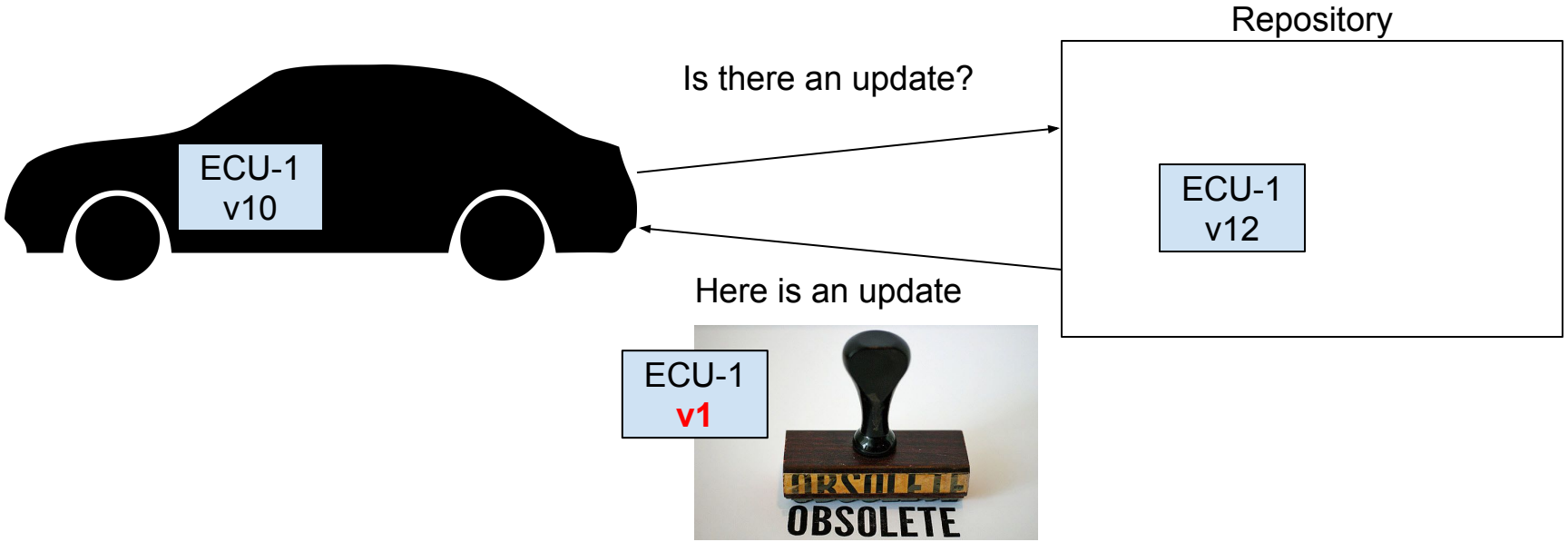
Arbitrary software attack



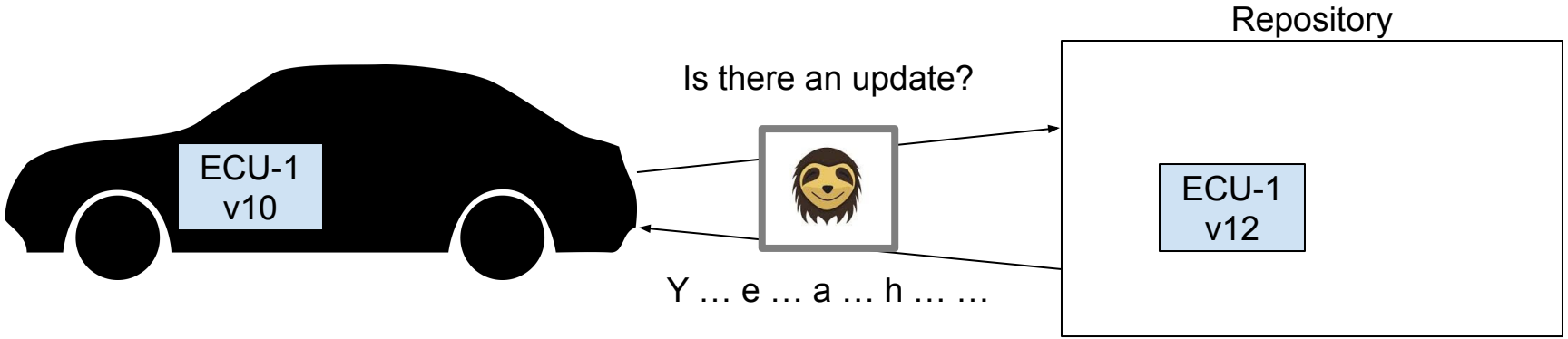
Freeze attack



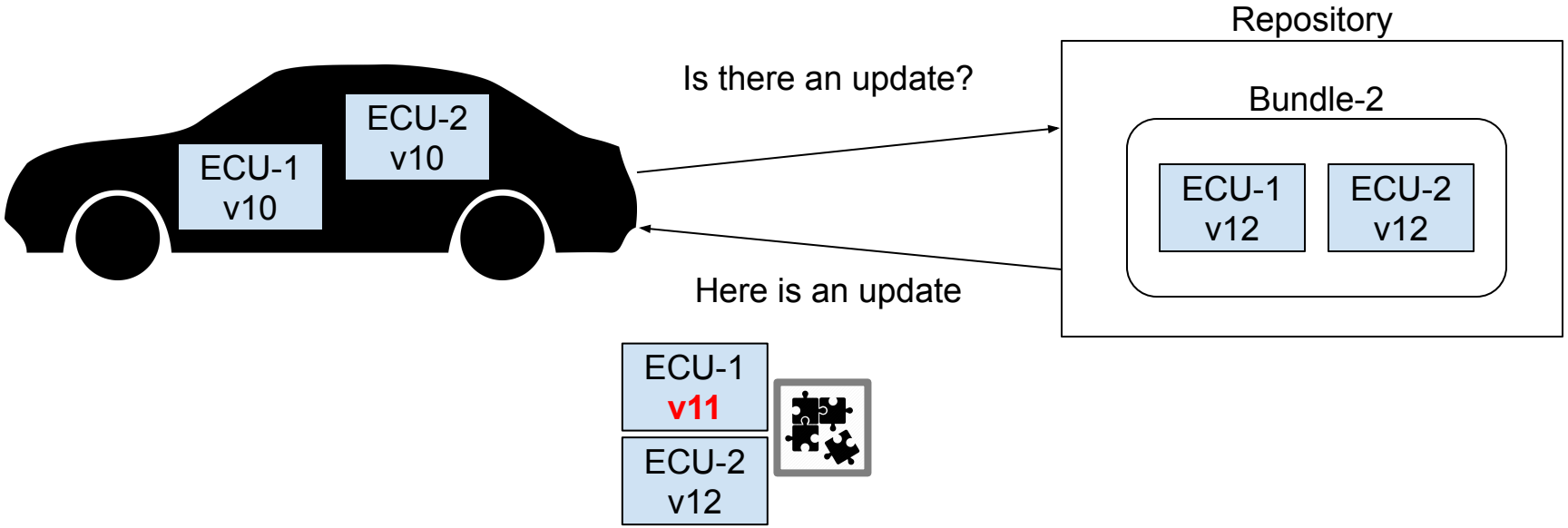
Rollback attack



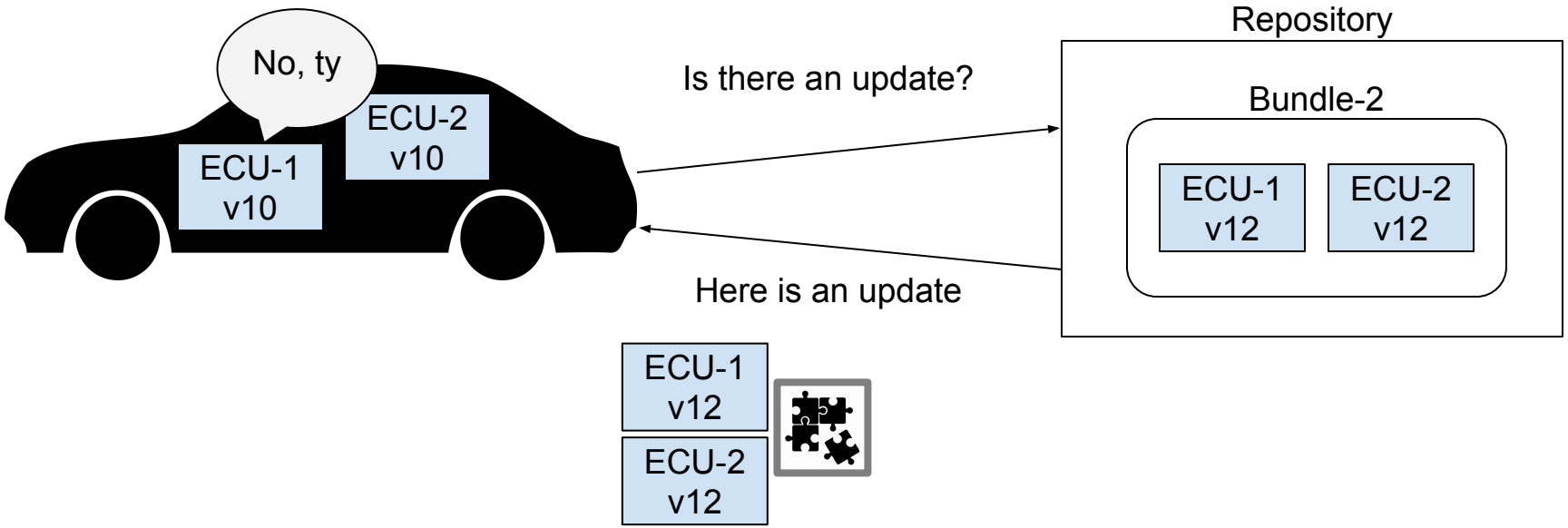
Slow retrieval attack



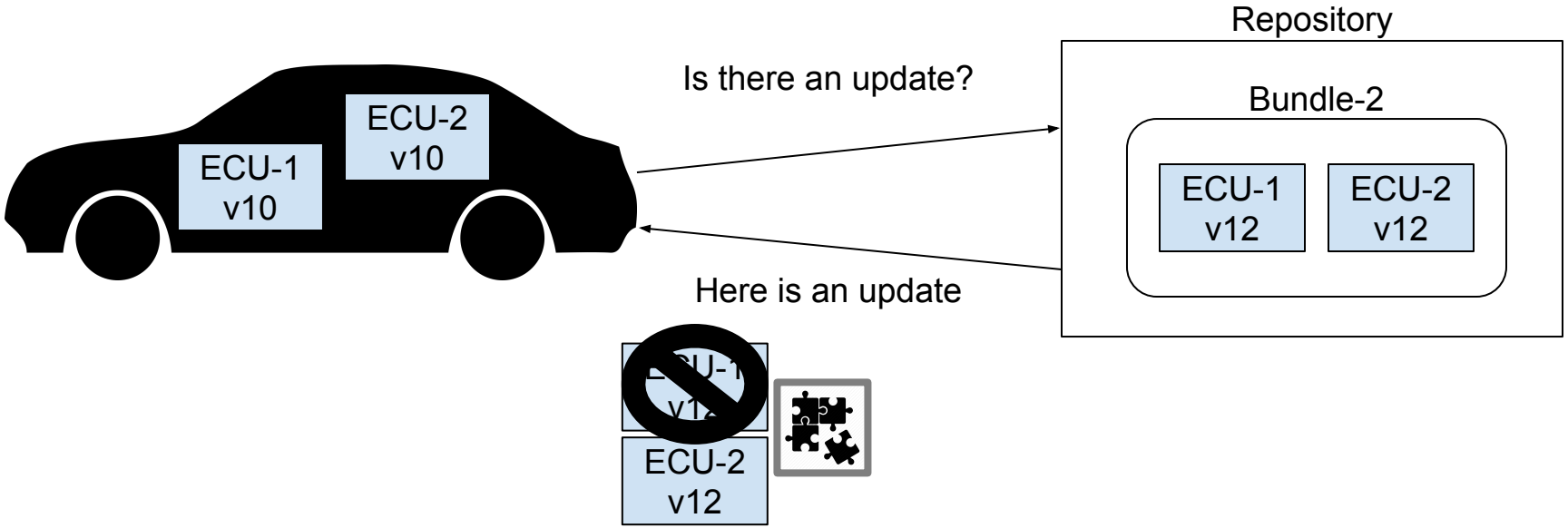
Mix and Match attacks



Partial Bundle attack

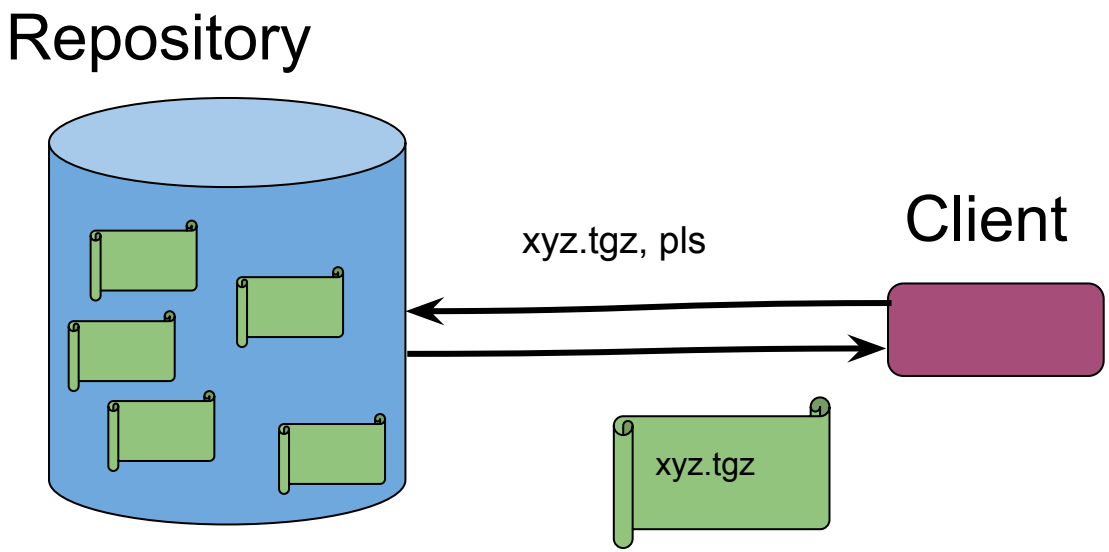


Partial Freeze attack



So how do people try to prevent these attacks?

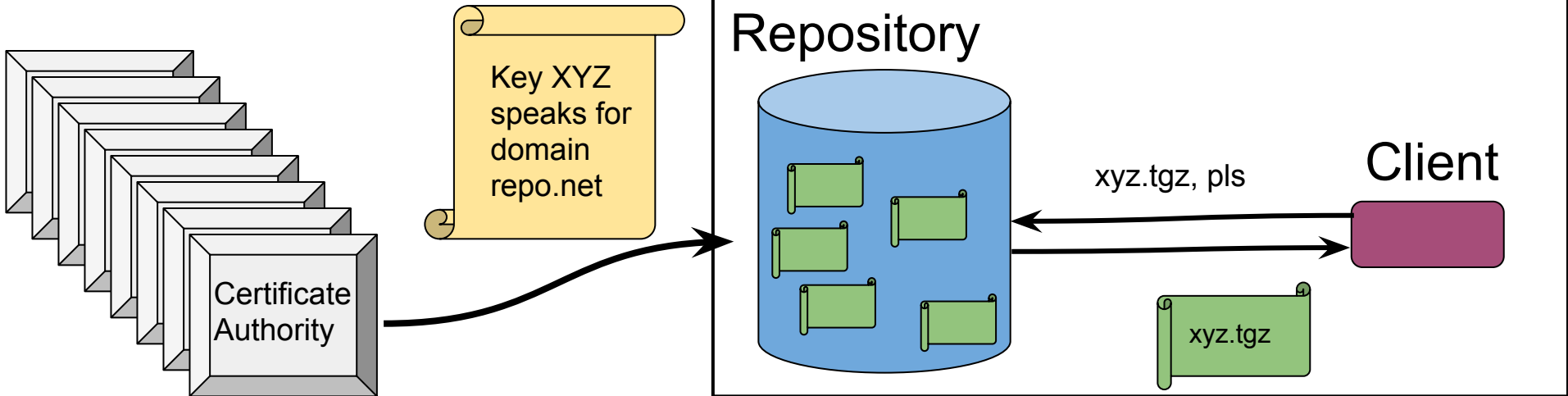
Update Basics



Inadequate Update Security 1: TLS/SSL

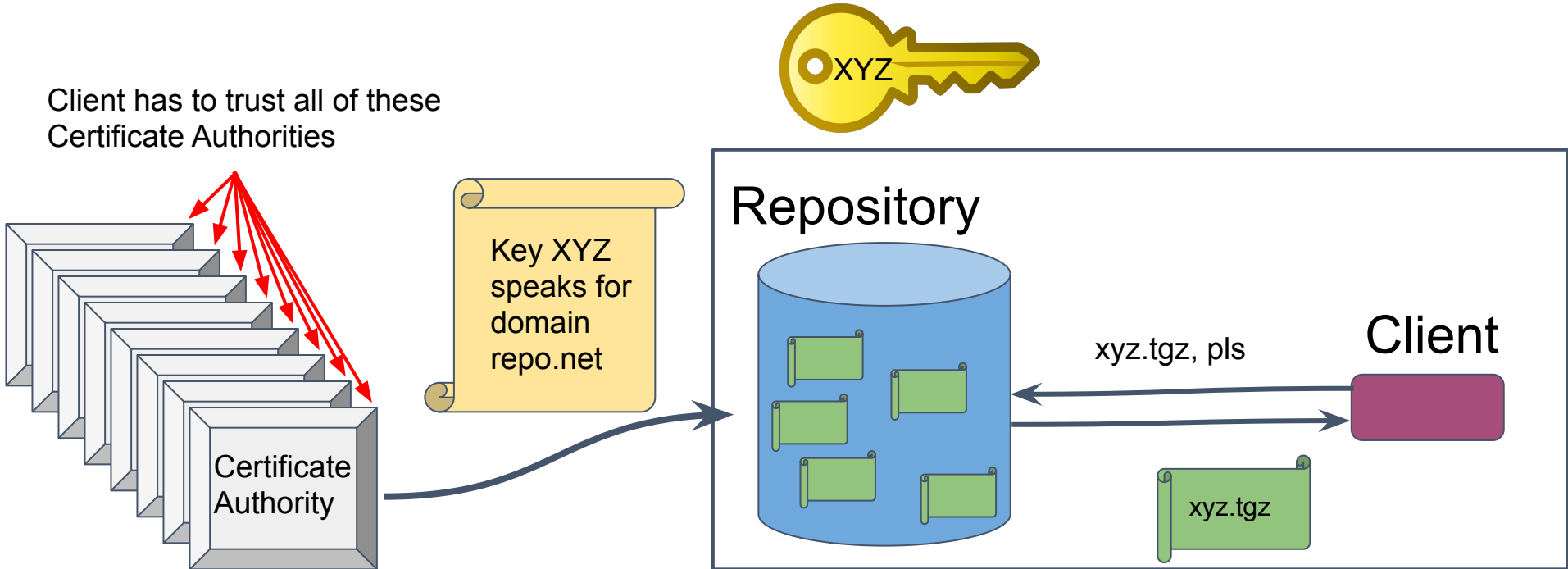
Traditional solution 1:

Authenticate the repository (TLS, SSL, etc)



Inadequate Update Security 2: TLS/SSL

Transport Layer Security: Problem 1

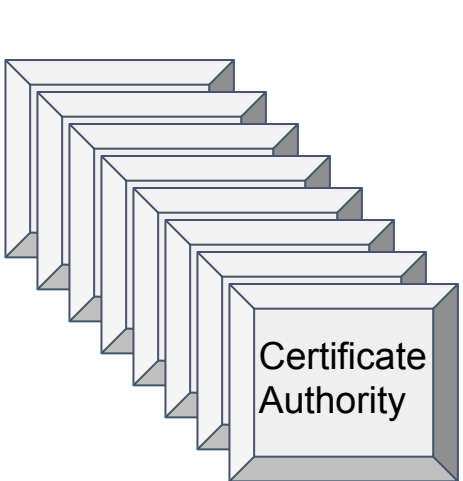


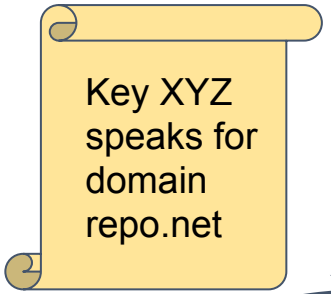
Inadequate Update Security 3: TLS/SSL

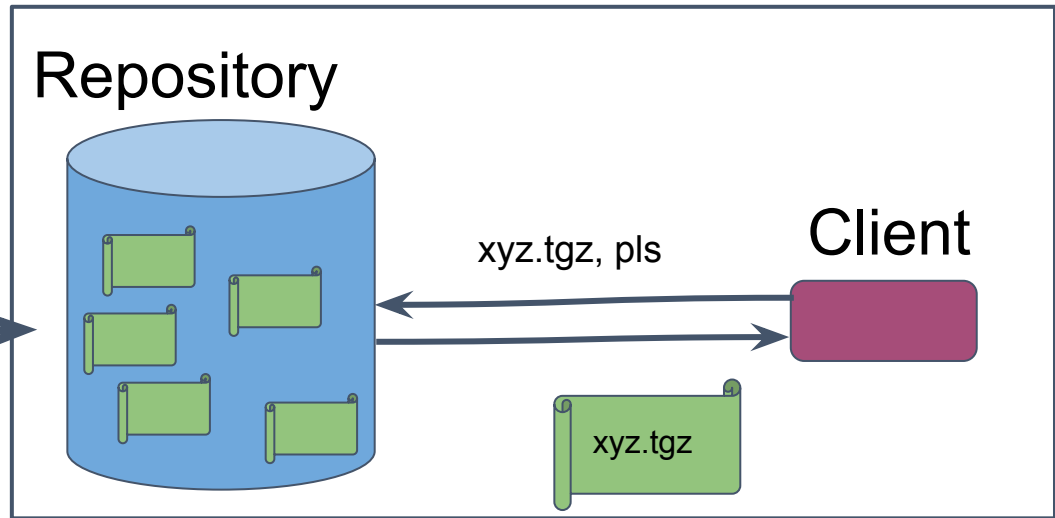
Transport Layer Security: Problem 2

Client has to trust this key.

… which HAS to exist ON the repository, to sign communications continuously.



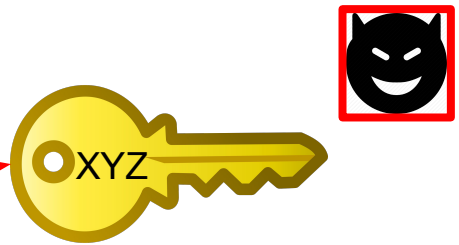




Inadequate Update Security 4: Just Sign!

Traditional Solution 2:

Sign your update package with a specific key.
Updater ships with corresponding public key.

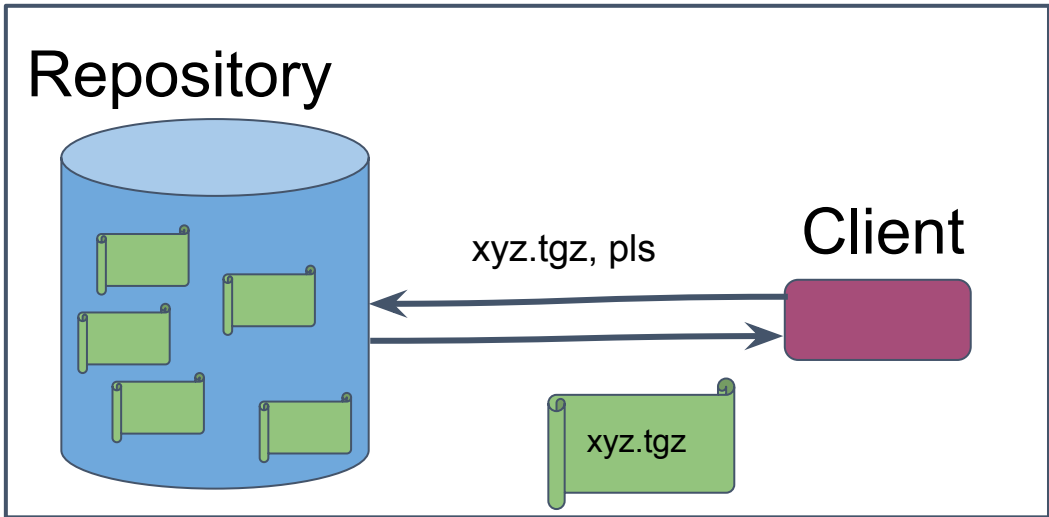


Client has to trust this key

… used for every update to the repository.

… key ends up on repo or build farm.

If an attacker gains the use of this key, they can install arbitrary code on any client.

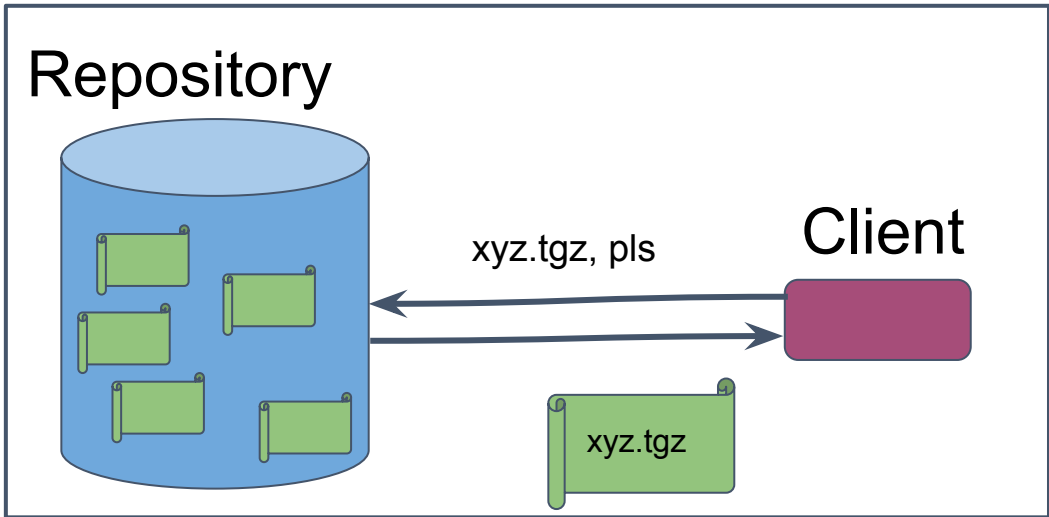


Update Security

We need:

- To survive server compromise with the minimum possible damage.
  - Avoid arbitrary package attacks
- Minimize damage of a single key being exposed
- Be able to revoke keys, maintaining trust
- Guarantee freshness to avoid freeze attacks
- Prevent mix and match attacks
- Prevent rollback attacks
- Prevent slow retrieval attacks
- ...

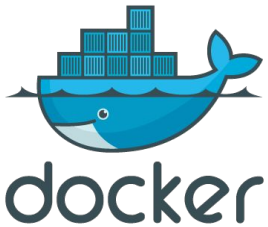
Must not have single point of failure!



The Update Framework (TUF)

Linux Foundation CNCF project

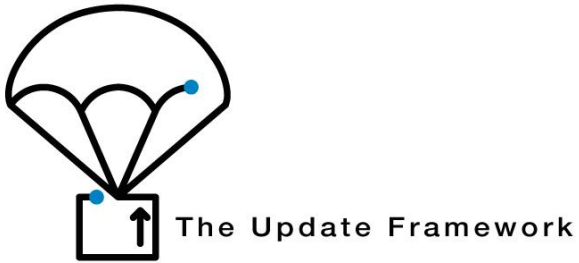
Widely used in industry:



The Update Framework (TUF): Goals

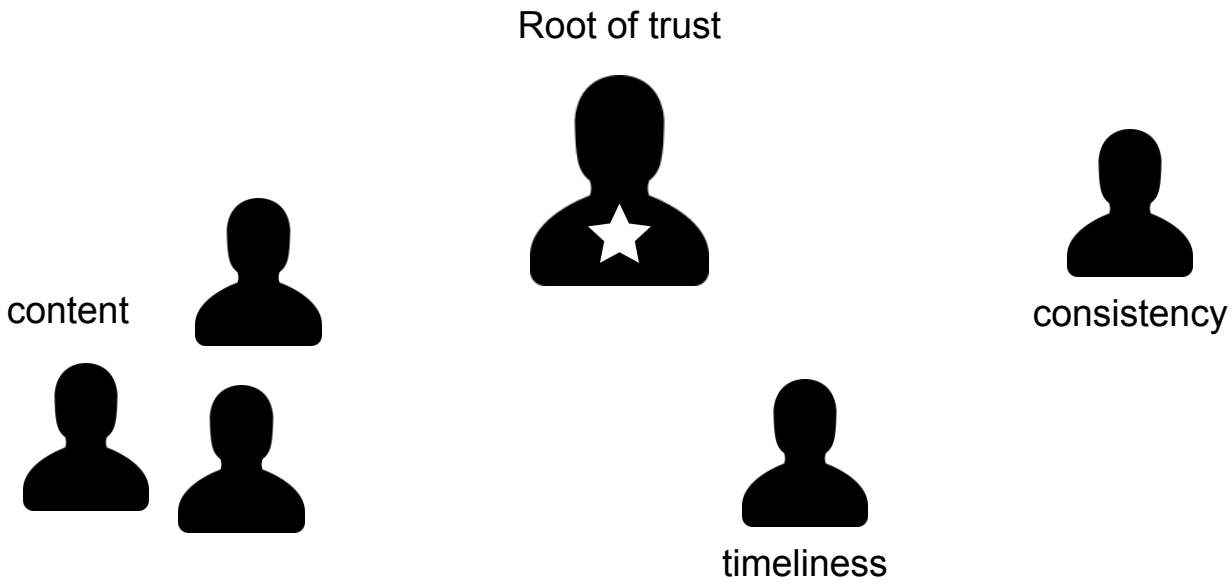
TUF goal “Compromise Resilience”

- TUF secures software update files
- TUF emerges from a serious threat model:
  - We do NOT assume that your servers are perfectly secure
  - Servers will be compromised
  - Keys will be stolen or used by attackers
  - TUF tries to minimize the impact of every compromise



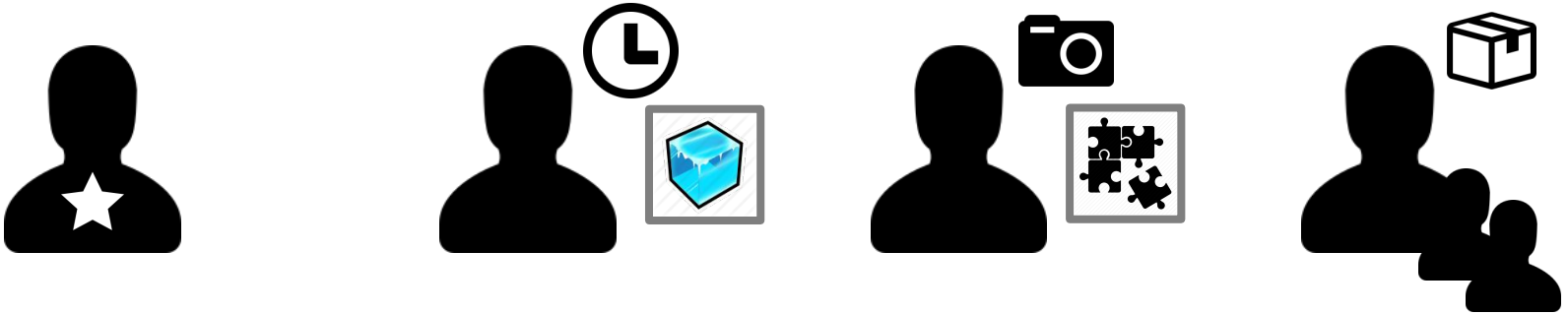
The Update Framework (TUF)

Responsibility Separation



The Update Framework (TUF)

TUF Roles Overview



Root

(root of trust)

Timestamps

(timeliness)

Snapshot

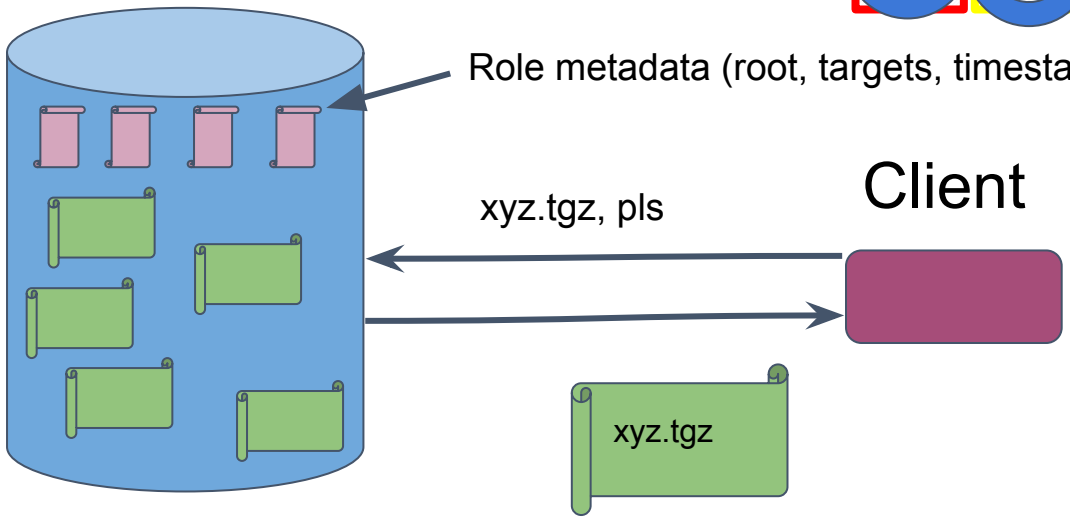
(consistency)

Targets

(integrity)

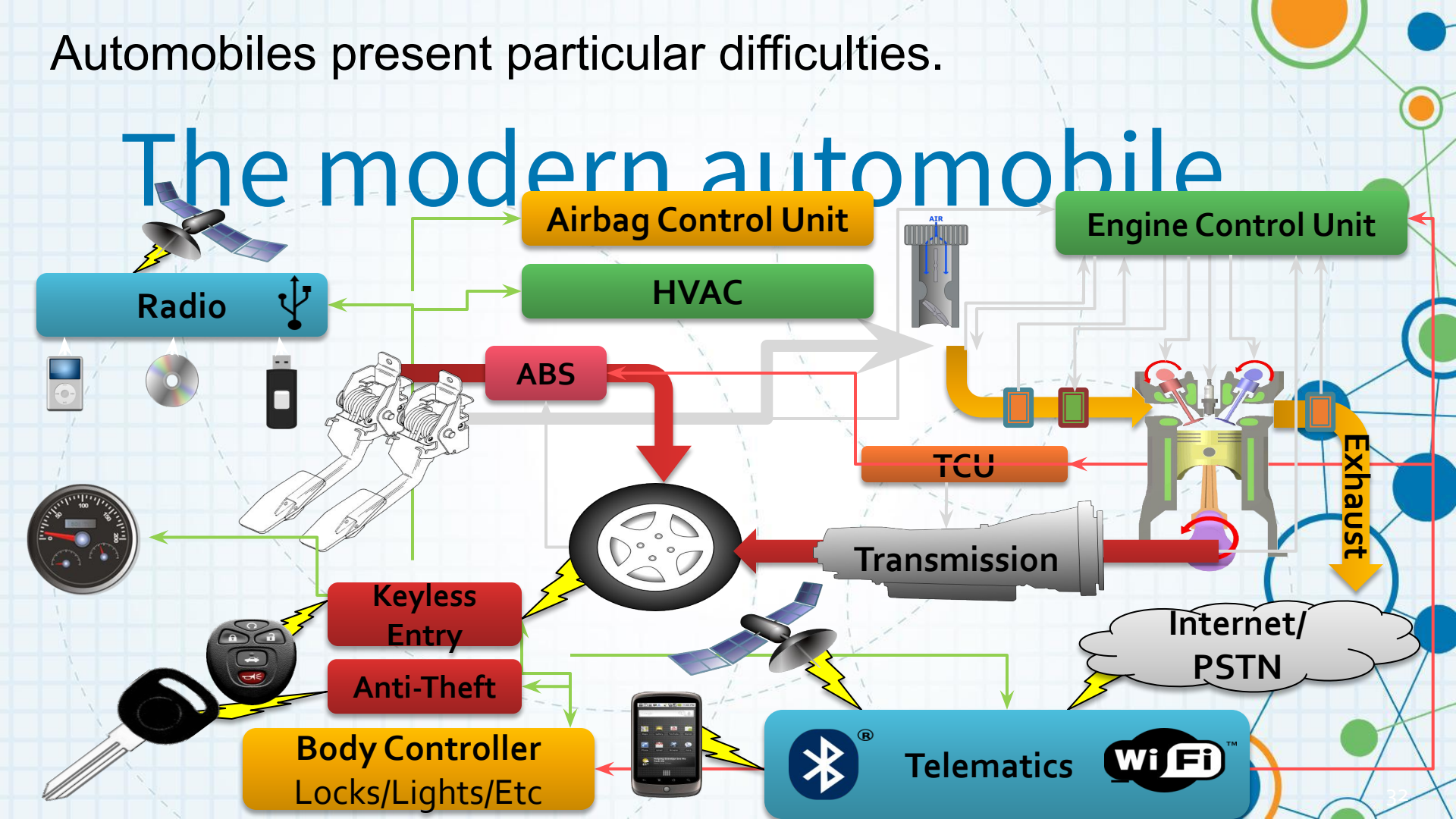
The Update Framework (TUF)

Repository



Automobiles present particular difficulties.

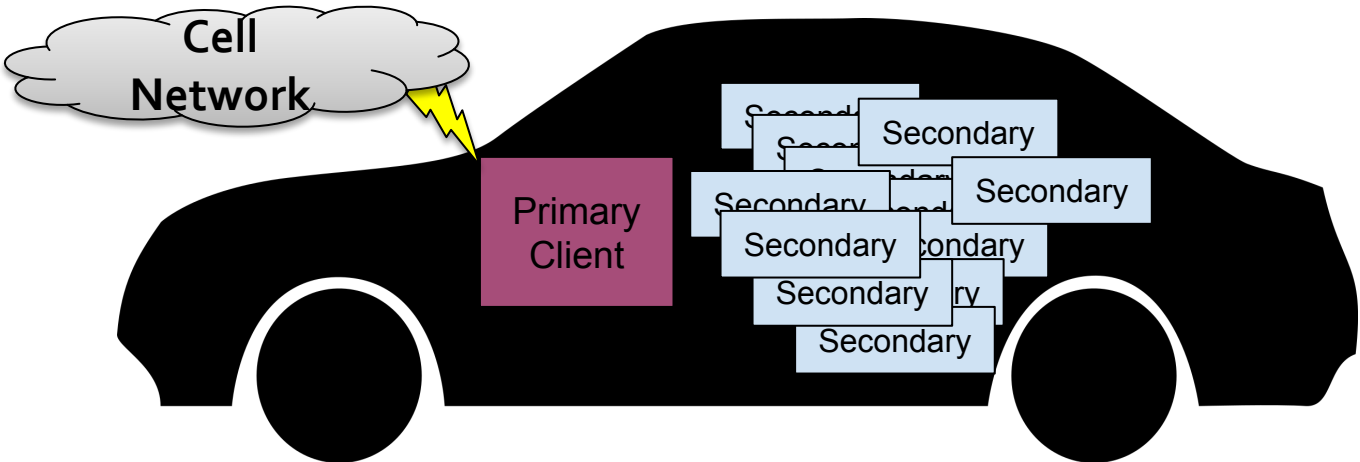
The modern automobile



Uptane builds on The Update Framework (TUF)

- Multiple Repositories: Director and Image Repository
- Manifests
- Primary and Secondary clients
- Full and Partial verification

Uptane: Client-side Basics



Uptane: High level view

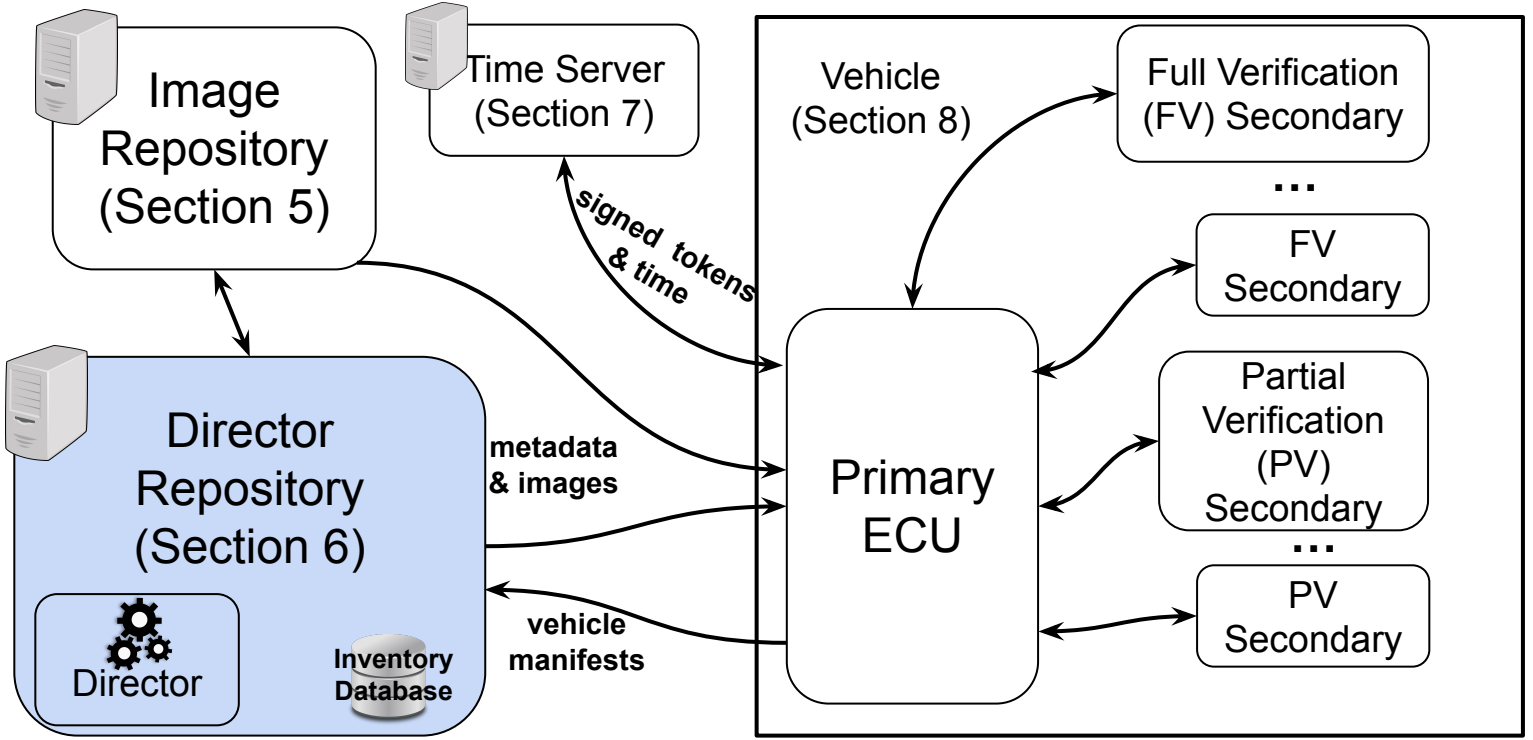
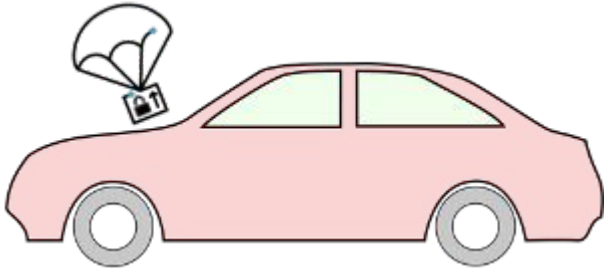
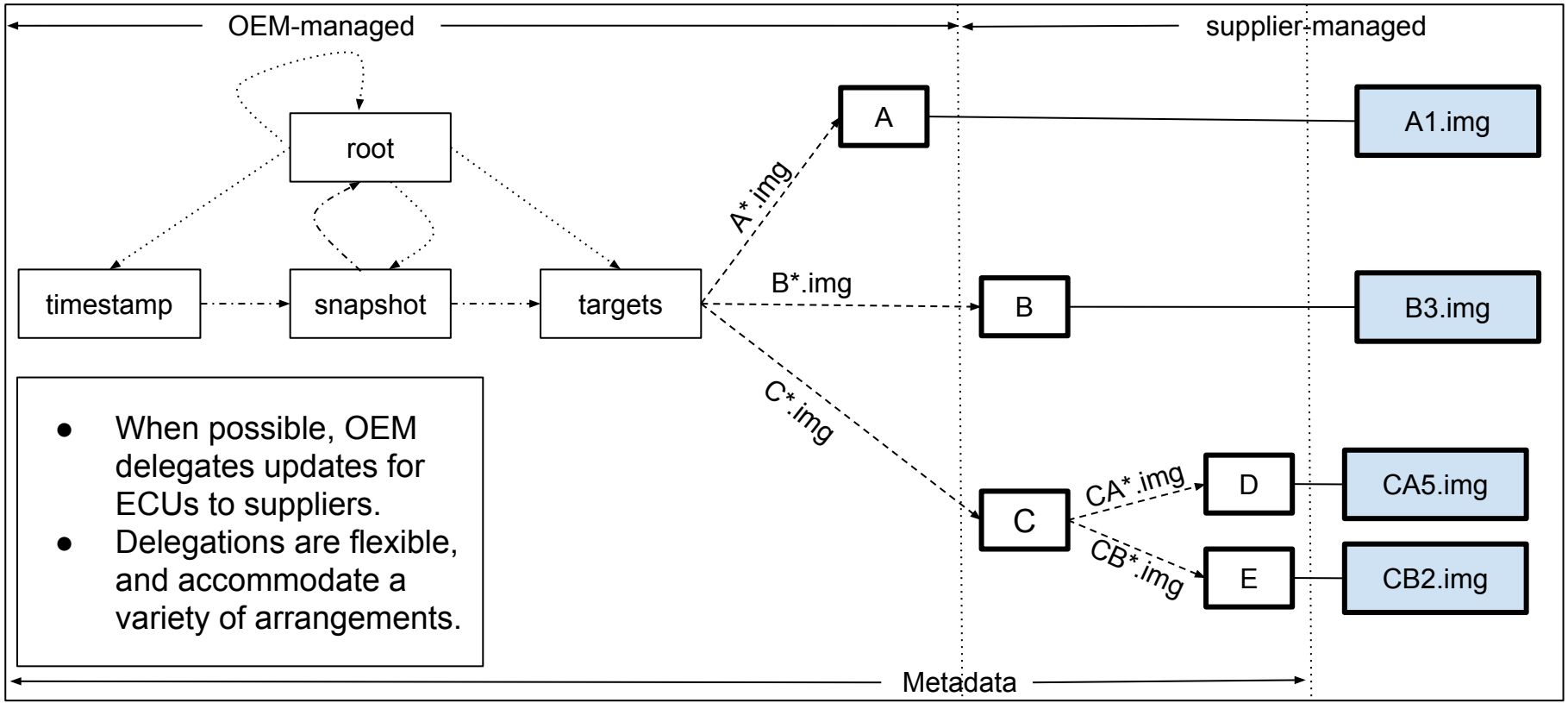


Image repository

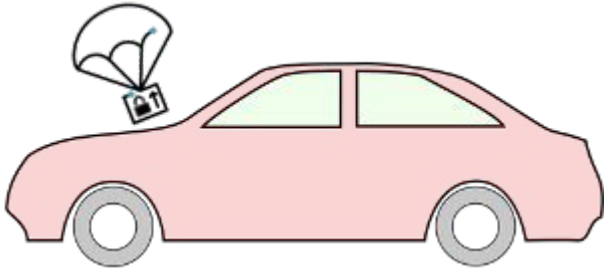


The image repository



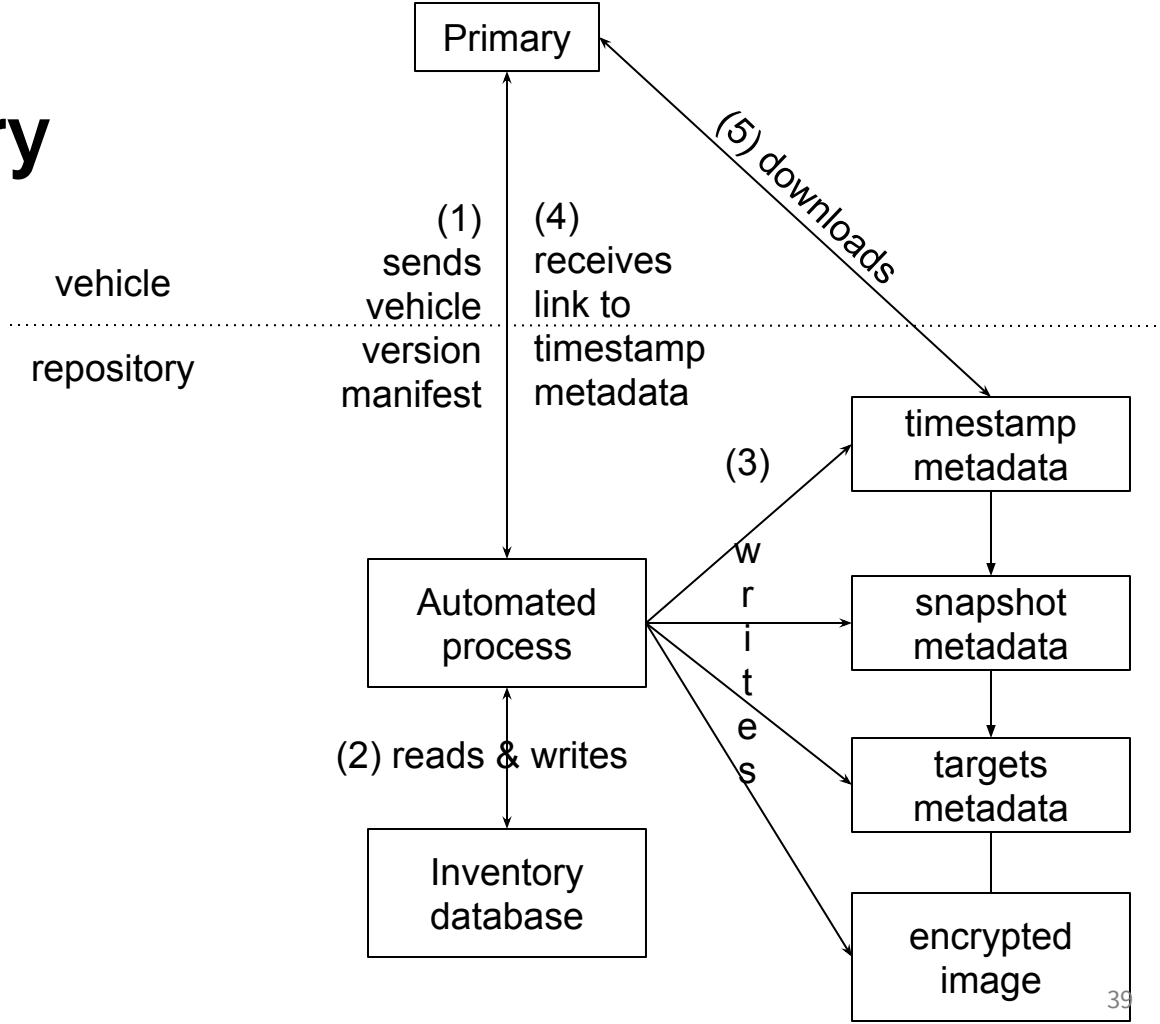
- When possible, OEM delegates updates for ECUs to suppliers.
- Delegations are flexible, and accommodate a variety of arrangements.

Director repository



Director repository

- Records vehicle version manifests.
- Determines which ECUs install which images.
- Produces different metadata for different vehicles.
- May encrypt images per ECU.
- Has access to an inventory database.



Big picture

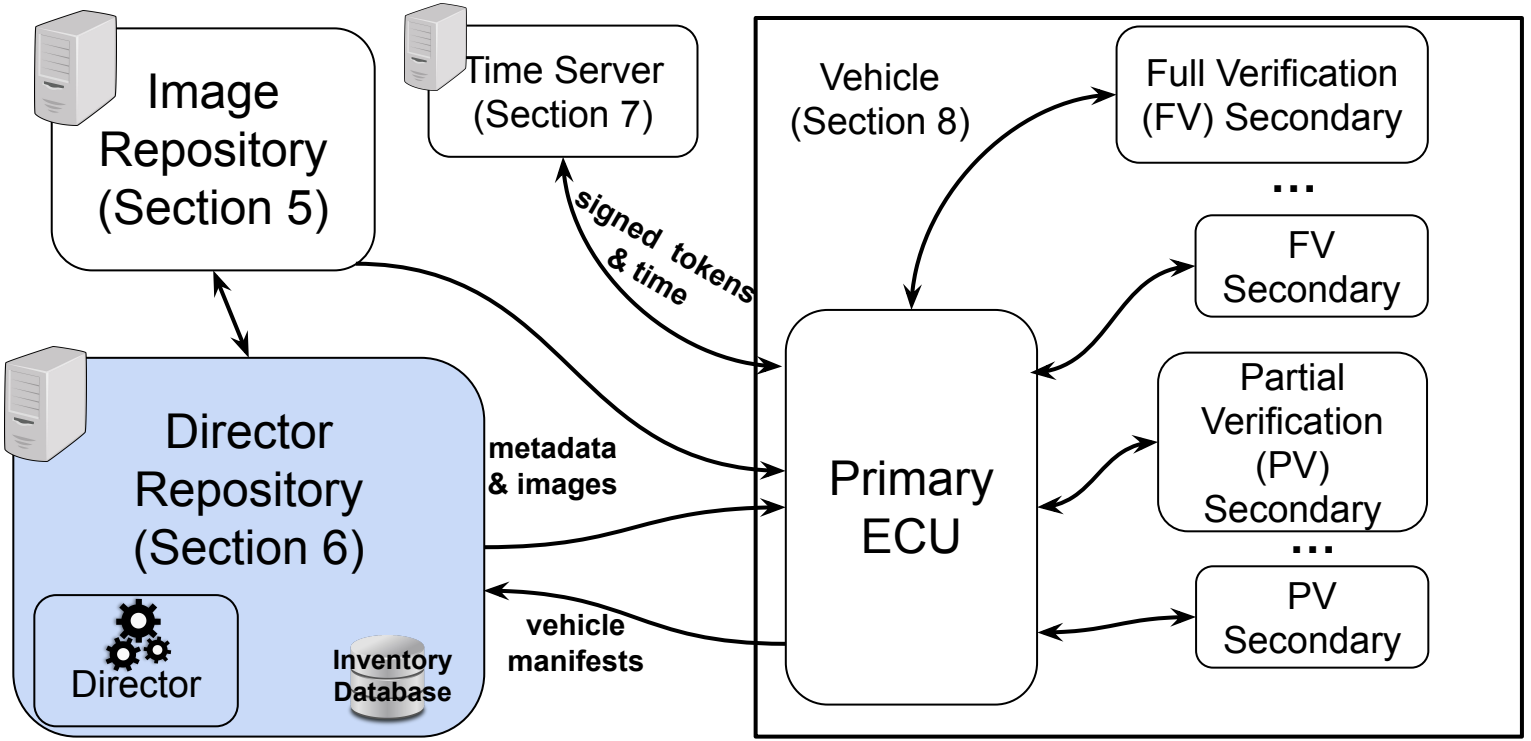
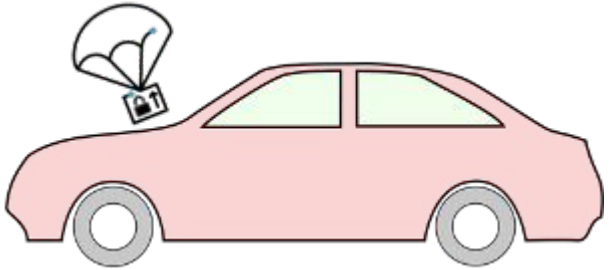


Image Repository (Section 5)
Time Server (Section 7)
signed tokens & time
Vehicle (Section 8)
Full Verification (FV) Secondary
...
FV Secondary
Partial Verification (PV) Secondary
...
PV Secondary
Primary ECU
Director Repository (Section 6)
metadata & images
vehicle manifests
Director
Inventory Database

Uptane status / wrap up



Uptane an Open and Secure SOTA system

- Multiple open source, free to use implementations
  - Diverse set of vendors and integrators
  - Other groups are free to contribute!

- Linux JDF standardization effort
  - Open for anyone to join (security reps from 78% of cars on US roads)
  - Other groups are free to contribute!
  - Free to join
  - Open and free specification

Security Reviews

Reviews of implementations and design:

- Cure53 audited ATS's Uptane implementation
- NCC Group audited Uptane's reference implementation (pre-TUF fork)
- SWRI provided Uptane reference implementation / specification audit
- ...

Uptane Integration

Work closely with vendors, OEMs, etc.

- Many top suppliers / vendors adopted Uptane in future cars!
    - Major OEMs in Europe, US, Asia
- Automotive Grade Linux
- OEM integrations
    - Easy to integrate!

AUTOMOTIVE GRADE LINUX

Press

- Dozens of articles
- TV / Radio / Newspapers / Magazines

POPULAR SCIENCE

WANT MORE?

TECHNOLOGY

The year's most important innovations in security

A botnet vaccine, a harder drive, and 3-D bag scanner.

By Kelsey D. Atherton and Rachel Feltman October 17, 2017

This article is a segment of 2017's Best of What's New list. For the complete tabulation of the year's most transformative products and discoveries, head right this way.

Year By

lligence Group BIG
d data
ard companies,
ce.

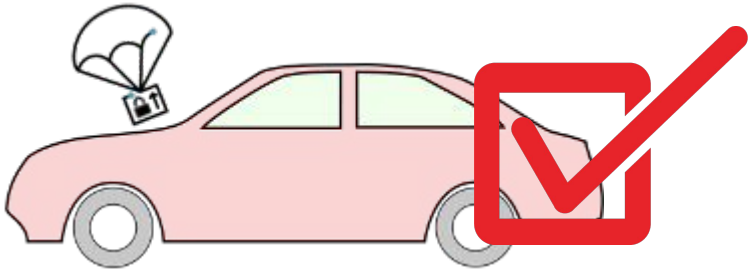
paigns-at scale-
with highly refined vehicle and device targeting, discrete policy and privacy controls, fully customizable consumer communications,
and solution deployment flexibility. In addition to the features announced in early 2017, OTAmatic now includes:

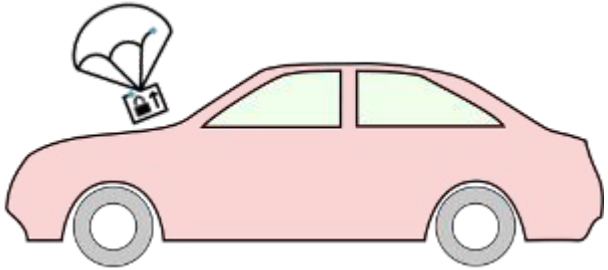
Get Involved With Uptane!

- Workshops
- Technology demonstration
- Compliance tests
- Standardization (Linux Foundation JDF)
- Join our community! (email: jcappos@nyu.edu or go to the Uptane forum)

https://uptane.github.io/

Homeland Security





For more details, please see the Uptane Standard at **https://uptane.github.io/uptane-standard/uptane-standard.html** and other documentation at **uptane.github.io**